Navigational Freespace Detection for Autonomous Driving in Fixed Routes

Narayan, Aparajit; Tuci, Elio; Sachiti, William; Parsons, Aaron

Published in:
ESANN 2020 - Proceedings

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Navigational Freespace Detection for Autonomous Driving in Fixed Routes

Dr. Aparajit Narayan[1], Dr. Elio Tuci[2], William Sachiti[1], and Aaron Parsons[1]

[1] Academy of Robotics, United Kingdom
[2] University of Namur, Belgium

Abstract. In this paper we propose a means for integrating vision-based road-detection and route planning in the context of autonomous driving in pre-defined routes. We train convolutional neural networks with three different input modalities to detect a triangular shape denoting drivable 'freespace' on the road which can also be used to interpret high-level navigational cues relating to turnings, junctions, intersections. These networks are developed for deployment in the control software of self-driving delivery cars being built by Academy of Robotics, a U.K based company which is looking to automate the last-mile delivery process. Networks are trained with raw camera inputs encoded in the RGB colour space, a hybrid colour space with no luminance channels and finally from an image composed from perception predictions of other deep neural network modules in the software pipeline. Results show that these networks even when trained with limited data are able to successfully learn/detect road-freepsace and adapt shape predictions appropriately when encountered with roundabouts, intersections etc. In addition, we propose means of integrating detections from networks corresponding to the three aforementioned input schemes through which further performance gains can be achieved.

Keywords: Road-Detection · Deep Learning · Autonomous Driving

1 Introduction

Road detection and navigating drivable areas on the road from camera inputs is a crucial aspect of autonomous driving. While there are a number of sub-fields dedicated to self-driving cars, the problem can almost be simplified to extracting 'freespace' areas on the input image plane which can then be the basis of navigational commands. This has been the subject of study for a number of decades and a wide body of work is dedicated to developing vision and/or control systems that can achieve this. The major challenges in the way of achieving this are due to the fact that real-world conditions exhibit a great degree of environmental variation with regards to lighting, road-structure, traffic etc.

This work is sponsored by Academy of Robotics, a UK based company aiming to develop autonomous delivery vehicles/drones. One of the main focus areas of our research is navigation in 'last-mile roads'. 'Last-mile' delivery in the logistics industry refers to the exponentially increasing costs of transporting a package from a depot/warehouse to the customers residence. This involves traversing residential, sub-urban environments where driving behaviour is different from long stretches of high-

way/motorway conditions. Our company aims to automate this particular aspect of delivery with the development of vehicles that robustly navigate such conditions. Designing generalized autonomous-driving software networks occupies the majority of the research and market space. However, we adopt a 'terrain-train' strategy wherein operational routes are pre-defined and the networks/algorithms comprising the software stack are trained and calibrated accordingly to 'specialize' in them.

1.1 Objective and Background

In this work we propose a novel means of road detection for autonomous driving in a scenario where the vehicle is required to operate in a fixed route. In such a case prior knowledge of high-level navigational decisions pertaining to driving in roundabouts, junctions or other unique features of the route is available. We can approach the problem of vision based road detection from the perspective of not only denoting 'driveable' space on the road but also generating cues through which intersections, junctions etc. and specific local features of the route can be navigated autonomously.

Traditionally, other academic studies have sought to solve issues related to road and lane detection as only demarcating which areas of the image correspond to the 'road', with route-planning and mapping considered separately. An exhaustive review of road detection/following is beyond the remit of this work. We shall highlight important methodologies, advancements and relevant prior research on this subject by the lead authors of this work. Within the wider field, some works are limited to detecting lane boundaries in structured environments such as highway roads. Initial approaches using manually specified filters for edge-extraction, were erorr-prone due to shadows, vehicle occlusion and sections where the lane marking disappeared due to weathering. Current state of the art works such as [5] have moved towards the use of deep convolutional networks which are better suited for handling the aforementioned issues.

When one considers unmarked paths, sub-urban/residential roads the detection problem gets greatly compounded. A number of works approach this by devising that can represent the road and/or non-road areas of the image plane. [8] successfully demonstrates autonomous driving in large stretches of unmarked desert roads using this principle. Here the road is modelled as a mixture of RGB gaussians and this 'road model' is updated with pixels from new frames to maintain adaptability on the course. Another example of successful autonomous driving using an adaptive road-modelling method can be found in [6], where the authors explore using a variety of colour models beyond the standard RGB. In this work new pixels are classified based on their relative Mahalonobis distance to the road colour distribution. A major factor influencing these techniques is the choice of features used to build the road/non-road models. Our observations from testing such methods have shown that in complex and varied operational conditions, a fixed set of features may not always be accurate in representing the road.

Deep Convolutional Networks with their ability to learn a robust hierarchy of features offer a more attractive solution to this issue of detecting complex road scenes as shown in [2] and [1]. Using the methodology outlined in [3] we were able to successfully demonstrate real-world driving using a deep CNN which detected unmarked, delineated roads. In another previous project ([4]), we used the principles of active-vision, embodiment and artificial evolution to generate a smaller sized recurrent network that

could directly control a mobile robot. This network controller was also able to navigate a robot in real-world environments much different from the virtual roads it was 'evolved in. The main draw-back of such neural network-based methods is their relatively lower operational accuracy in environments much different from the datasets/simulations they were trained in. Moreover, the issue of navigating junctions/intersections and exhibiting behaviour specific to certain sections of the route (one-way vs two-way) remains unaddressed by them.

Prior to the development presented in this paper, we created a manual design-based algorithm/module for road-freespace detection. The specific implementation details of this shall not be presented here but in short, the objective of this software was to divide the image plane into a grids and predict which grid could be classified as being 'free' (or drivable). This algorithm processed output frames from three neural networks in our software pipeline. More details on these networks, each of which achieves a specific function (object-detection, scene segmentation, lane detection) can be found in section 2.3. Examples of the freespace detection using this approach can be seen in figure 1.

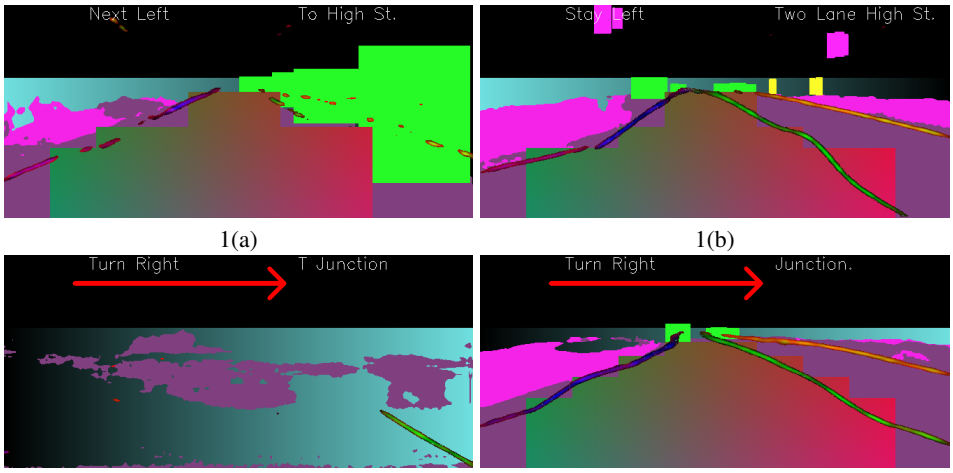


Fig. 1. Examples of the 'intermediate' images that are processed by our rule based algorithm along with the corresponding freespace shape. The predicted shape composed of detecting individual pixel grids can be seen in images 1(a), 1(b) and 2(b) as the shaded block-like region in the middle. Image 2(c) is an example of no detectable freespace from the intermediate image.

1.2 Motivation

Whilst being successful in extracting drivable areas using the aforementioned 'intermediate' image as a input there were still a number of limitations which led us to the line of research detailed in this paper. One of the main limitations was that this is still

a somewhat rule-based method constrained by the assumptions made during the design process. Therefore, there are some instances when the lower level networks exhibit incorrect predictions and these errors are carried through to the final freespace detection. A recurrent example observed by us in our test datasets was in cases of extreme lighting (bright spots mixed with shadows) which had an effect on the scene segmentation network. Moreover, the lane detection module also incorrectly labelled edges which were not lanes or road boundaries in some frames. A counter to this could be to simply better fine-tune the lower level networks on the chosen route. Safety is however of paramount importance for commercial self-driving applications and we need to supplement this freespace detection with a system that can be more robust to such types of noise. Another argument for implementing an alternate freespace module was that the detections from the previous design-based method still needed to be supplemented with GPS data to make navigational decisions during turns, junctions, roundabouts etc. GPS cannot be relied upon for precision and even if the approximate current location is known it is still challenging to implement a set of rules that can manage a vehicle during these parts of the route. It may be better to approach this problem by implementing a neural network that can indicate upcoming 'high-level navigational decisions' from the vision input.

2 Methods

We aim to develop a convolutional neural network based road-freespcae predictor which is also able to intrinsically provide information for mapping/route-management related decisions. This work is a comparative study wherein we explore generating mid-sized convolution networks for this task of 'navigational freespace' detection. These networks have the same architecture but use different input modalities/schemes. Subsequently in section 2.1 we elaborate the baseline method using a standard RGB image. Here we also detail the road-freespace shape and network architecture which is common for all networks. In sections 2 and 3 we explain two alternate input schemes; first using a hybrid colour model devoid of luminance and the second using detections of other neural networks. We also present another methodology in section 2.4 where we propose combining freespace detections of multiple CNNs using a regression model as a potential means to achieve improved accuracy. Section 2.5 details features of the route dataset where these networks are trained and tested. Note that while it is an attractive concept in theory to train a single network and explore if it displays better accuracy with a different input scheme/modality; previous works by this team arrived at the conclusion that deep neural networks tend to work best when the same input scheme that was used during training is presented for testing. Therefore, we train separate neural networks, each corresponding to a particular input scheme.

2.1 Baseline Method: RGB

For the first network we feed the raw camera image encoded in the standard RGB (red, green, blue) as the network input after resizing it to the required shape of 336 x 152 (W x H). The current deep convolutional network architecture shown in figure 2 was fixed after a period of experimentation with architectures of different depths and width in

each layer. We also experimented with implementing 3-D convolutional networks which require a multi-frame input rather than a single frame. However, this approach did not yield satisfactory results and we shifted to implementing standard 2-D convolutional neural networks with input dimensions being $Height * Width * Channels*$. The three channels of the input image depend on the particular input modality being used. Another consideration for affixing the network architecture and not adding further convolution layers is the execution speed, especially considering this is meant to be deployed in a real-time software pipeline and will be one of the several networks that need to run their update cycle multiple times per second.

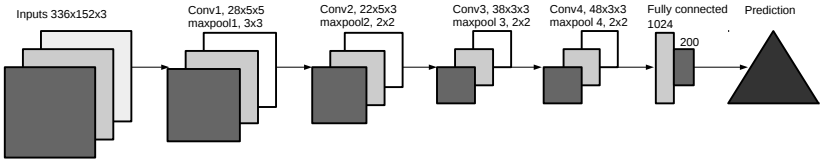
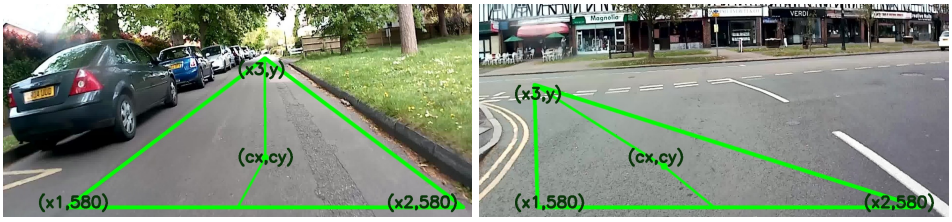


Fig. 2. Diagram illustrating the CNN architecture used for these experiments.

Freespace Shape Model The triangular road freespace shape is generated by four floating point values predicted by a convolutional neural network. These values **x1**, **x2**, **x3**, **y** are explained in figure 3 with regards to their role in composing the final triangular shape. We acknowledged that there are other possible shapes that can be generated with 4 floating point values which may intuitively better suited for detecting road space. However, the focus of our work is more on demonstrating the ability of such networks to learn map specific cues and incorporating this into the detection of freespace on the road. The triangular road shape is the latest iteration in our attempts at exploring a suitable model for this particular type of route specific road detection. We were unsuccessful in experiments to train networks that were supposed to output a classification of grids (road and non-road) on the image plane. The intention was to directly augment this with the same grid-based classification processed using the design based algorithm which is part of our current software pipeline (see section 1.1). Without physical on-vehicle trials it is difficult to develop and calibrate a robust control system for translating features/predictions generated by the perception software into steering/acceleration signals. Keeping this in mind we herein propose a simple mechanism which demonstrates the ability of this freespace shape (when predicted accurately) to be the basis of generating desired navigational directives. As shown in figure 3, assuming the middle column of pixels corresponds to the vehicle pointing straight, a simple trajectory can be generated if two goal points of the centroid and tip of the triangular freespace model are set. While the freespace shape is enough to provide steering and speed/acceleration signals on its own, it is likely to be integrated with signals from other modules/algorithms within an overall decision structure when deployed in our autonomous vehicle.



1(a) 1(b)

Fig. 3. These two images show examples of the triangular freespace shape being overlaid on the image plane. Standard image dimensions for our software are 1344 (W) x 608 (H). The three vertices of the triangular shape have the following coordinates (x1,580), (x2,580), (x3,y); out of which x1, x2, x3 and y are predicted values. The triangle centroid is shown by the label (cx,cy) in these images. In 1(a) the vehicle is required to make a sharp left and stay on the left lane of this junction. In 1(b) the vehicle needs to steer slightly right to avoid coming near the parked cars and then maintain a straight course. Note that origin coordinates (0, 0) correspond to the top leftmost pixel.

2.2 Hybrid Colour Model: HSA

One of the drawbacks of the RGB colour space is that colour and brightness are both represented in a single value; i.e. the level of a particular channel (R, G or B) denotes how much colour as well how much 'brightness' is present. This may not be ideal for developing robustness to extreme lighting conditions. Therefore, we train a network with an alternate hybrid colour space with no channels for luminosity. We use 'Hue (H)' and 'Saturation (S)' from the HSV colour space and the '*a' channel from the L*a*b colour space to create a hybrid representation hereon referred to as 'HSA'. Refer to figure 4 for visualizing images encoded as 'HSA'.

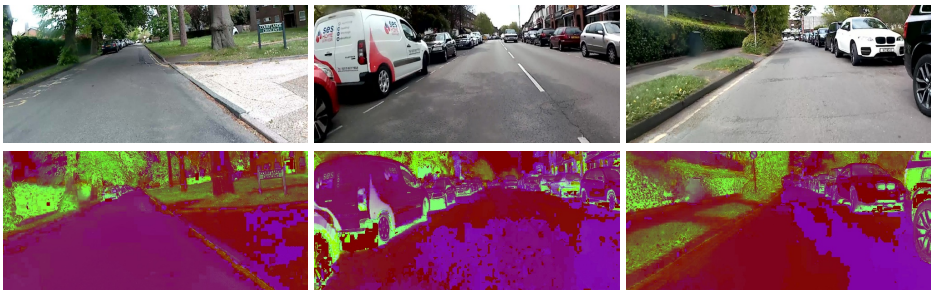


Fig. 4. The top row shows raw example frames from the dataset. Below these in the second raw is visualization of the corresponding raw image above, when encoded in the 'HSA' colour-scheme. For the second-row images the red channel is substituted with '*a' from 'L*a*b', green with 'Saturation' from 'HSV' and blue with 'Hue' from 'HSV'. All channels are normalized to the 0-255 range.

2.3 PERCEPT: Object Detect, Segment, Lane Detection

The third network for navigational freespace is trained on images that are formed by fusing the outputs of three other convolutional neural networks, each performing specific functions (see figure 5). This is inspired by our prior attempts to develop a design-based freespace detection algorithm by processing the detections of these three networks (see section 1.1 and figure 1). Features crucial to autonomous driving such as obstacles, lane boundaries, road and pavement areas are already explicitly highlighted in this input scheme. A network using this type of input image may be more adept at detecting 'navigational freespace' in noisy environments as these cues will always be visible to some degree. There are instances where the output frames of the object, lane and segmentation networks are also noisy. By representing such scenarios in the training set, the freespace network can be immune to these errors. This may not the case if a designed rule-based algorithm is used, as errors in the initial network detections would reflect strongly in the final freespace shape. Provided below are details of the three networks whose outputs are fused to form the input image.

- **Object Detection/Classification:** This is a implementation of the 'YOLOV3' network in the darknet framework, the details of which can be found in [7]. Currently this is one of the best performing networks for general purpose object detection/classification in the field. The network outputs bounding box coordinates for every object detected in the input frame which are then translated to opaque rectangles of different colours on the frame. General objects (traffic lights, road sign etc) are coloured as pink; cars, trucks, buses are marked blue, two-wheelers including cycles are coloured red and pedestrians are marked yellow.
- **Scene Segmentation:** This is a implementation of an end-to-end pixel wise scemantic segmentation network using the ICNET architecture, details of which can be found in [9]. Each pixel is coloured as one of 18 colours according to the 'CITYSCAPES' dataset labelling convention. Pixels for the 'road' class are classified as purple.
- **Lane Detection:** This is a deep neural network designed for performing end-to-end detection of lane boundaries from a raw input image. The design features of this network are detailed in [5]. The network outputs a final frame with coloured pixels predicting as belonging to the lane and black pixels for all other areas.

Initial attempts involved feeding the prediction frame of each network as a separate channel to the network. The learning was found to be quite poor using this scheme and therefore we feed the intermediate image encoded in RGB colour space as the network input. It should also be noted that none of these networks are fine-tuned for the specific route as we want to explore the ability for the freespace prediction network to robust to noise in the detections provided by the prior networks. The final deployment version will feature networks fine-tuned on images from the designated operational area. The network corresponding to this input scheme shall hereon be referred to as 'PERCEPT' for the remainder of this paper.

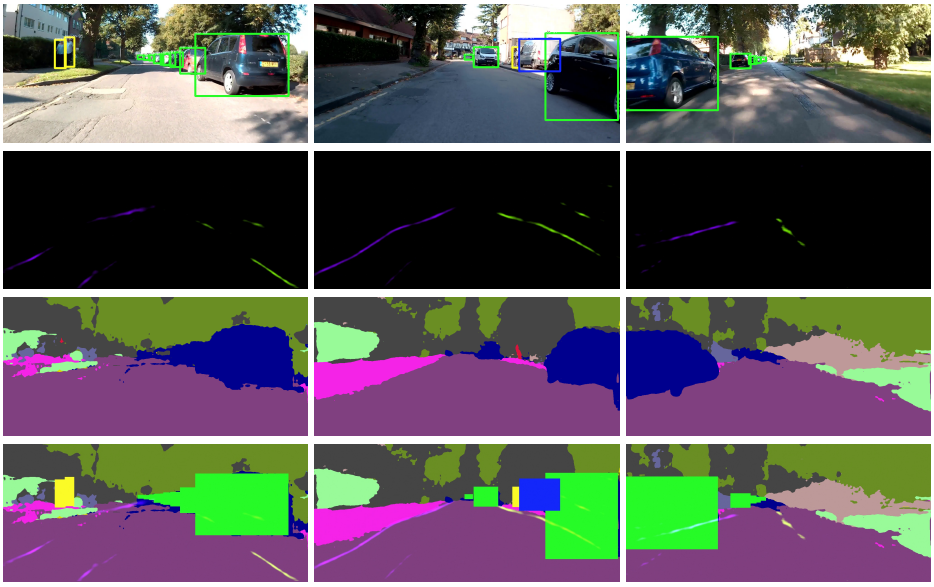


Fig. 5. Figure showing individual predicted frames of the object-detection, lane detection and segmentation networks, as well as the final fused image which forms the input for the freespace network. The topmost row shows examples of the object-detection network predicting bounding boxes around cars and pedestrians. The second from top row shows the output of lane-detection on the corresponding frames above. The third from top row contains segmentation network outputs. The bottom row shows the fnal image that is formed after fusing all three network outputs.

2.4 Mix: Combining multiple network predictions

Besides presenting detection results related to these networks individually, we also explore 'navigational freespace' detections if the shape predictions of the three networks are to be combined using different techniques. It should be noted that our deployment software stack is meant to be executed on 6 camera inputs on 2 Nvidia-Drive PX2 computers. Real-time operation is a prime consideration and tests have shown that we can execute these three networks simultaneously given our computational infrastructure and that these freespace networks are not very large architectures, having only 4 convolution layers. We implement five techniques for combining/mixing the three network detections. The first is a simple averaging scheme. Given every network predicts 4 floating point numbers, we simply average the corresponding parameters of all three networks to give 4 final values which translate to the triangular shape. The other 4 are different regression schemes implemented using the python scikit-learn library. By name these multi-output regression modles are **k-nearest neighbour** regression, **random forest** regression, **decision tree** regression (with the depth set at 8) and **gradientboost** regression. They are 'fit' on the predictions of the three networks against the ground-truth annotation on 1000 images from the training dataset. It could be argued that fitting these regression models on test set images could have produced a better fit, however due to limited testing data we have to use images from the training sequence loops for

now. Details on each of these particular regression models can be found in *https://scikit-learn.org/stable/index.html*.

2.5 Route Dataset

We recorded 5 videos of a GoPro camera mounted on a car (in the same configuration, height) driving around a fixed route from start to finish. These drives, carried out in different dates and time of the day were meant to capture the environmental variation that is present within the same route of road. Variation with regards to traffic (some videos were captured during peak hours), lighting conditions, presence of pedestrians etc. were observed in this compilation of videos. Frames for each video were extracted at 4 fps providing 5 datasets with 1341, 1276, 1299, 1205, 1038 frames respectively. The drives took place in a mostly residential area of Surrey, London and involve the vehicle being driven from a fixed starting point, exiting the residential area which is a one lane road onto the high-street which is a two-lane road, going 360 degrees on a round-abound and returning to the starting-point travelling in the opposite direction. The route also features two junctions where the correct turn needs to be made on the way forth and back. The residential sections of the road have restricted space with rows of cars often parked on the side. On-road traffic increases significantly after the second turn onto the high-street. Refer to figure 6 for a map overview of the route trajectory. 3 of these datasets/image-sequences were used for training and in section 3 we present results of their testing on two sequences with 1205 (testset 1) and 1038 (testset 2) frames respectively.

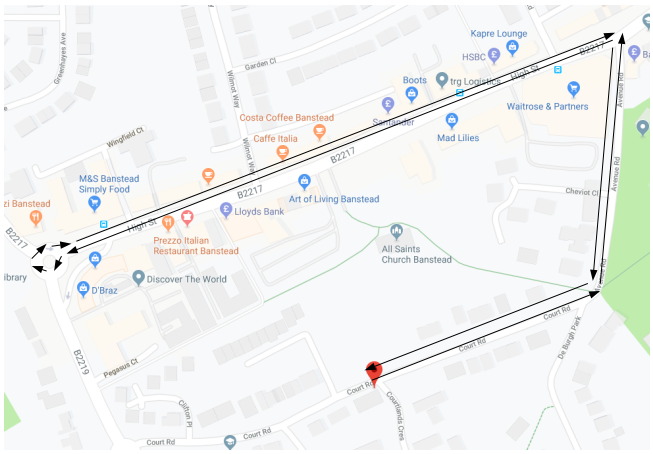


Fig. 6. Image showing a map-level overview of the route. Arrows indicate the direction of travel. Red beacon marks the start/finish point. The entire loop takes approximately 8-10 minutes to drive.

3 Results

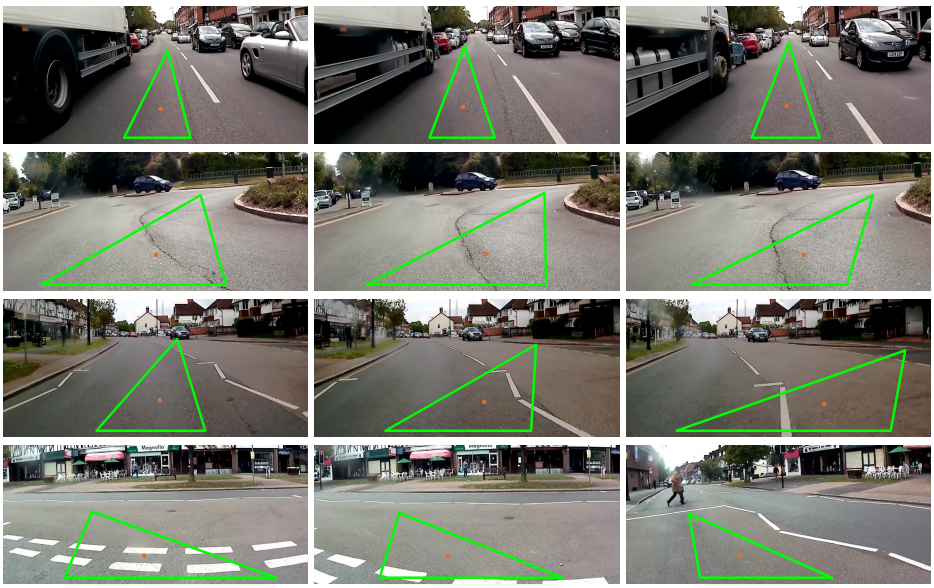


Fig. 7. Examples of the triangular shape prediction being overlayed on four image sequences. The predictions are examples of the 'PERCEPT' network. It should be noted that the shape derived from the other two networks (RGB, HSA) and those from the combination of all three also exhibited similar behaviour. In the sequence (left to right) shown in topmost row the vehicle is in a two-lane road and needs to stay on the current lane while being surrounded by traffic on both sides/directions. The network correctly predicts a straight, narrow triangle keeping sufficient distance away from the truck and cars. In the second from top sequence the vehicle requires to make a U-Turn on the upcoming round-about. The network steadily shifts the triangle base to the left whilst also shifting the vertices to suggest going right. In the third row from top the vehicle is required to exit the current lane and take a sharp right turn. The network recognizes this portion of the route and changes the triangular shape to denote this turn even though the vehicle orientation seems to keep moving straight. In the bottom sequence there is a need to take a sharp left at this junction and stay on the left lane. This is again correctly predicted by the low triangular shape (with the tip pointing left) thereby giving no space for driving straight.

The results detailed in this section are form freespace predictions of the three networks detailed in section 2.2 (RGB, HSA, PERCEPT), and further experiments where the detections of these individual networks were combined through the different schemes detailed in section 2.4. Statistics for these combined or mixed predictions detailed in tables 1 and 2 are presented with the prefix 'MIX' and the post-fix labelling the particular combination scheme used (avg, knn, rf, dtree, gradient). Error statistics are calculated from the comparison of the triangular shape generated from these predictions against the human ground-truth annotation. These human ground-truth annotations can be con-

sidered noisy as it was left up to the annotator to subjectively define the triangular freespace with only a few rules set by us to adhere.

From the 8 detection sequences (3 individual, 5 combined) for each of the 2 test-sets it could be observed that in almost all frames the projected freespce shape could accuractely capture drivable areas of the road. The networks were also successful in learning to avoid including portions of the sidewalk, curb, pedestrians and other cars in the freespace. There were some frames for the networks with HSA and RGB input channels (when tested individually) where small portions of the projected triangle included non-drivable areas. These were however limited and it could be observed that the networks made adjustments in subsequent frames to bring the detection back within drivable space (even though each frame is treaded independently). Moreover it was observed that the PERCEPT network was especially adept at keeping the detection away from such non-drivable areas and there were no instances where it could be observed that the detection would lead the vehicle onto another car, pedestrian or onto the wrong lane. This was also true for the scenarios where predictions from all three networks were combined using regression methods.

For all cases the predicted freespace triangles exhibit desired detection behaviour in special sections of the route related to junctions, sharp turns, lane changes etc. They were able to differentiate between areas where the road was a 2-way drive (in the high street) and kept detection within the desired lane even though there was space available in the adjoining lane. Another characteristic observed was that the networks could adjust to dynamic movement of obstacles (cars, pedestrians) in subsequent frames and change the trajectory downwards/sideways to keep away from there. Analysis of sample detection frames for the PERCEPT network displaying the characteristics discussed above can be found in Figure 7. Table 1 shows the median and standard deviation of

Networks	**Test 1**				**Test 2**			
	Centroid°		**Tip°**		**Centroid°**		**Tip°**	
	med	*std*	*med*	*std*	*med*	*std*	*med*	*std*
RGB	-5.7	24.9	1.2	20.8	-1.3	20.9	-4.9	17.8
HSA	-5.9	29.3	2.8	22.1	3.4	24.3	-2.1	25.6
PERCEPT	-4.5	23.5	1.5	19.1	1.0	19.0	-2.1	15.5
MIX(avg)	-5.8	22.5	1.8	15.3	0.0	18.0	-2.7	14.3
MIX(knn)	-2.1	19.5	0.4	12.3	2.2	18.6	-0.4	14.3
MIX(rf)	-2.5	20.7	2.3	13.7	2.4	17.9	-1.4	13.9
MIX(dtree)	-1.5	27.3	0.9	17.5	-0.7	16.3	-0.4	13.8
MIX(gradient)	-2.4	18.0	2.0	14.4	2.6	15.5	-1.3	12.6

Table 1. Table showing median (med), standard deviation (std) of errors relating to centroid and tip angles (in degrees) to for both test sets. Centroid angle refers to the angle generated by a vector arising from point [672 (x), 580 (y)] in a 1344x608 image to the freespcae triangle centroid. Tip angle refers to the angle generated by a vector arising from the freespace triangle centroid to its tip formed by predicted parameters (x3, y). Refer to figure 3 for a better understanding of these vectors. For reference we use a pixel coordinate system where the top-left pixel is (0,0).

errors for the two test image sequences/datasets (see section 2.5) for two parameters (centroid angle and tip angle) that are inferred from the predicted triangle shape. In section 2.1 we make the case of generating steering/trajectory commands on the basis of

these two angles. Analysing the error statistics of these angles in table 1, for individual networks 'PERCEPT' is marginally more accurate than 'RGB' and 'HSA'. Accuracy also improves when a combination of all three networks is employed using the regression models. In all cases errors for the tip angle are relatively lower than the centroid angle. This is attributed to the networks being more accurate or consistent with the ground-truth in predicting the top-vertex (x3, y) values than x1 and x2 which mark the base of the freespace triangle (see figure 3). A major feature of the predicted triangle sequences was even though within drivable road areas the predicted shape do not always match the manual annotated ground truth shape. Indeed, the predicted freespace shapes, irrespective of the network and/or combination scheme were generally narrower at the base compared to annotated triangles. This contributes to the higher standard deviation errors for the tip angle parameter in table 1. Some examples of the discrepancy between ground-truth and predicted freespace shapes can be seen in figure 8.

Networks	**Test 1**				**Test 2**			
	overlay %		**only out %**		**overlay %**		**only out %**	
	med	*std*	*med*	*std*	*med*	*std*	*med*	*std*
RGB	19.2	25.1	49.6	36.2	12.3	23.5	43.3	34.1
HSA	15.2	24.7	58.1	36.6	15.3	23.8	57.3	36.2
PERCEPT	21.3	25.7	38.1	35.8	24.2	24.5	28.1	33.2
MIX(avg)	21.8	22.7	27.3	34.7	21.9	20.9	21.4	32.6
MIX(knn)	24.5	25.7	16.1	34.3	24.8	25.3	15.0	32.5
MIX(rf)	17.9	19.3	14.9	35.7	24.3	19.2	10.6	31.6
MIX(dtree)	26.2	26.0	17.8	36.5	27.3	19.9	10.5	34.3
MIX(gradient)	21.7	21.0	14.7	34.9	26.8	19.6	10.8	32.0

Table 2. Table showing median and standard deviation of two parameters which analyse the degree of overlap between the predicted and ground-truth triangles for both tests. The column named 'overlay' refers to percentage of overlap between the two said triangles. 'only out' refers to the percentage of predicted triangle that lies outside the ground-truth area. RGB, HSA and PERCEPT are the three trained networks described in section 2.2. The remaining rows with the prefix MIX display results with the predictions of the three aformentioned networks combined using different techniques (see section 2.4).

Table 2 shows results from comparing the area overlap between the predicted and ground-truth freespace shapes. We devise two parameters for this measure. These are, 'overlay' which is the amount of overlap between the two shapes and 'only lab' which is percentage of the predicted triangles outside the ground-truth triangle. It can be generally inferred from these statistics (combined with visual examination of frame sequences) that out of the three individual networks, the PERCEPT network performed better when it comes to not including non-annotated areas in their freespace projection. In other words, the predicted freespace triangles for this network stay within the area of the ground-truth to a greater degree than the other two. The results also seem to suggest that there is merit in combining the detections of the three individual networks using the regression approach detailed in section 2.4. This led to generally less error rates and in none of the frames belonging to these sequences could we observe the projection including non-traversable areas.

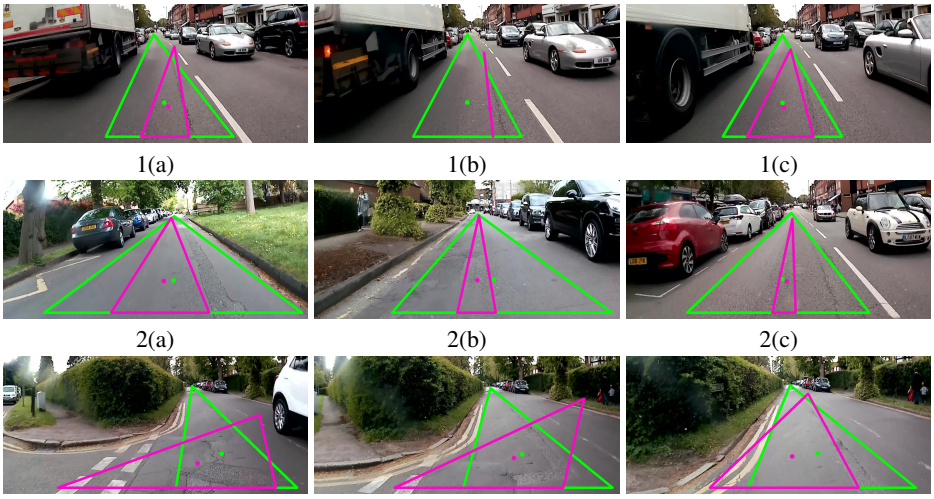


Fig. 8. Examples of the triangular shape prediction being overlayed on three image sequences. The predictions are examples of the 'PERCEPT' network shown as pink triangles and the manual ground-truth annotations as green triangles. It should be noted that the shapes derived from the other two networks (RGB, HSA) and those from the combination of all three also exhibited similar behaviour. In images 1 (a, b, b) we can see the vehicle needing to travel through a narrow section of the road whilst being obstructed by a lorry/truck on the left and an incoming car on the opposite lane to the right. The manual triangle annotation in green maintains a generally straight triangle with the perimeter being closer to the non drivable space. The pink predicted shape is much narrower and in case of 1 (c) narrows down to almost a line. The top vertex however remains close to that of the annotated triangle. Images 2(a, b, c) are three unconnected frames showing instances of the predicted triangular shape being much narrower contributing to larger errors for x1, x2 when compared to x3 and y (the top vertex coordinates). Images 3(a,b, c) show an instance where the vehicle is required to travel on the right branch of the junction with a car approaching from the opposite direction. There is a mismatch between annotated and predicted shapes in frames 2(a) and 2(b), although it could be argued that the predicted shape is still projected within 'road space'.

4 Conclusion and Future Work

The methodologies presented in this work have laid the foundation for an interesting avenue of research which can provide a novel means for translating camera inputs to directional cues that integrate mapping/route-planning information. Despite limitations arising from limited training images and noisy annotation, all three individual networks trained were found to be generally capable of learning road 'freespace' i.e include only road areas in their projected shapes as well as appropriately alter the freespace predictions in junctions and roundabouts. Within these individual networks the 'PERCEPT' network was found to be more robust and less prone to including areas beyond the ground-truth adding weight to our method of using fused perception outputs of other neural networks as the input image. We also observe that combining the freespace pre-

dictions of these individual networks via simplistic regression models increase the performance. The final triangle prediction arising from this regression-based combination scheme negates errors that may be present in individual network detections.

Our focus for upcoming works is to explore the use of these projected triangle shapes to control a real autonomous platform. We are especially interested in replacing the regression method with a recurrent neural network that can use sequences of freespace triangles generated by one or multiple such convolutional networks to provide a final triangular shape. This final freespace shape can be integrated with the design based freespace algorithm discussed earlier (section 1.1) to provide steering and speed commands for the vehicle. A parallel line of research is to have a recurrent neural network using these predicted freespacce shapes to directly predict control outputs (speed, steering). While such a scheme may not be possible when using the full resolution image, using only sequences of multiple freespace triangles may offer an effective means of dimensionality reduction for a recurrent neural network to directly control an autonomous vehicle. In conclusion this work presents a novel approach of considering road-detection and shows that neural network models when trained with the right conditions and architecture can learn to predict freespace in a manner that can also direct an autonomous vehicle in areas where multiple trajectories are possible and/or a special type of detection behaviour is required.

References

1. Alvarez, J., Gevers, T., LeCun, Y., Lopez, A.: Road scene segmentation from a single image. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Proc. of the European Conf. on Computer Vision, pp. 376–389. Springer Berlin Heidelberg (2012)
2. Chen, C., Seff, A., Kornhauser, A., Xiao, J.: DeepDriving: Learning affordance for direct perception in autonomous driving. In: Proc. of the IEEE Int. Conf. on Computer Vision (ICCV). pp. 2722–2730 (2015)
3. Narayan, A., Tuci, E., Labrosse, F., Alkilabi, M.: Road detection using convolutional neural networks. In: Proc. of the 14^{th} European Conference on Artificial Life (ECAL). pp. 314–321 (2017)
4. Narayan, A., Tuci, E., Labrosse, F., Alkilabi, M.H.M.: A dynamic colour perception system for autonomous robot navigation on unmarked roads. Neurocomputing **275**, 2251 – 2263 (2018). https://doi.org/https://doi.org/10.1016/j.neucom.2017.11.008, http://www.sciencedirect.com/science/article/pii/S0925231217317228
5. Neven, D., Brabandere, B.D., Georgoulis, S., Proesmans, M., Gool, L.V.: Towards end-to-end lane detection: an instance segmentation approach. CoRR **abs/1802.05591** (2018), http://arxiv.org/abs/1802.05591
6. Ososinski, M., Labrosse, F.: Automatic driving on ill-defined roads: An adaptive, shape-constrained, color-based method. Journal of Field Robotics **32**(4), 504–533 (2015)
7. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018)
8. S. Thrun et al.: Stanley: The robot that won the DARPA grand challenge. Journal of Field Robotics **23**(9), 661–692 (2006)
9. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: Icnet for real-time semantic segmentation on high-resolution images. CoRR **abs/1704.08545** (2017), http://arxiv.org/abs/1704.08545